When Punctuation Matters: A Large-Scale Comparison of Prompt Robustness Methods for LLMs

Mikhail Seleznyov[1,2], **Mikhail Chaichuk**[1,3,6], Gleb Ershov[5], Alexander Panchenko[1,2], Elena Tutubalina[1,4,6], Oleg Somov[1,7]
AIRI[1], Skoltech[2], HSE University[3], Sber AI[4], Yandex[5], ISP RAS Research Center for Trusted AI[6], MIPT[7]

TL;DR: LLMs are highly sensitive to minor formatting changes. Prior research has addressed different aspects of prompt sensitivity, but there is a lack of systematic evaluation across tasks, models, and learning paradigms. We fill this gap by benchmarking 4 robustness methods in a unified framework across 3 LLM families and distribution shifts, and provide actionable takeaways for practitioners.

Motivation

LLMs are highly sensitive to minor format changes.
Many methods [1,2,3,4] have been developed to alleviate this issue, however:

- they have not been compared in a **unified setting**,
- their performance under **distribution shifts** is poorly understood,
- the impact of **sampling strategies** on format sensitivity remains poorly understood.

Methodology

Datasets:
- Natural Instructions (subset of 52 tasks), classification & multiple-choice.
- GSM8K-platinum, long-form mathematical generation

Methods:
- **Few-shot** (FS, baseline, standard in-context learning without any robustness-improvement techniques.)
- **Batch Calibration** [1] (BC, post-hoc adjustment of logits using token statistics from a batch.)
- **Template Ensembles** [2] (TE, averages predictions across multiple prompt formats.)
- **Sensitivity-Aware Decoding** [3] (SAD, penalizes high-variance token probabilities during decoding)
- **LoRA with augmentations** (LoRA, finetuning with LoRA adapters using prompts with varied capitalization, separators, and spacing)

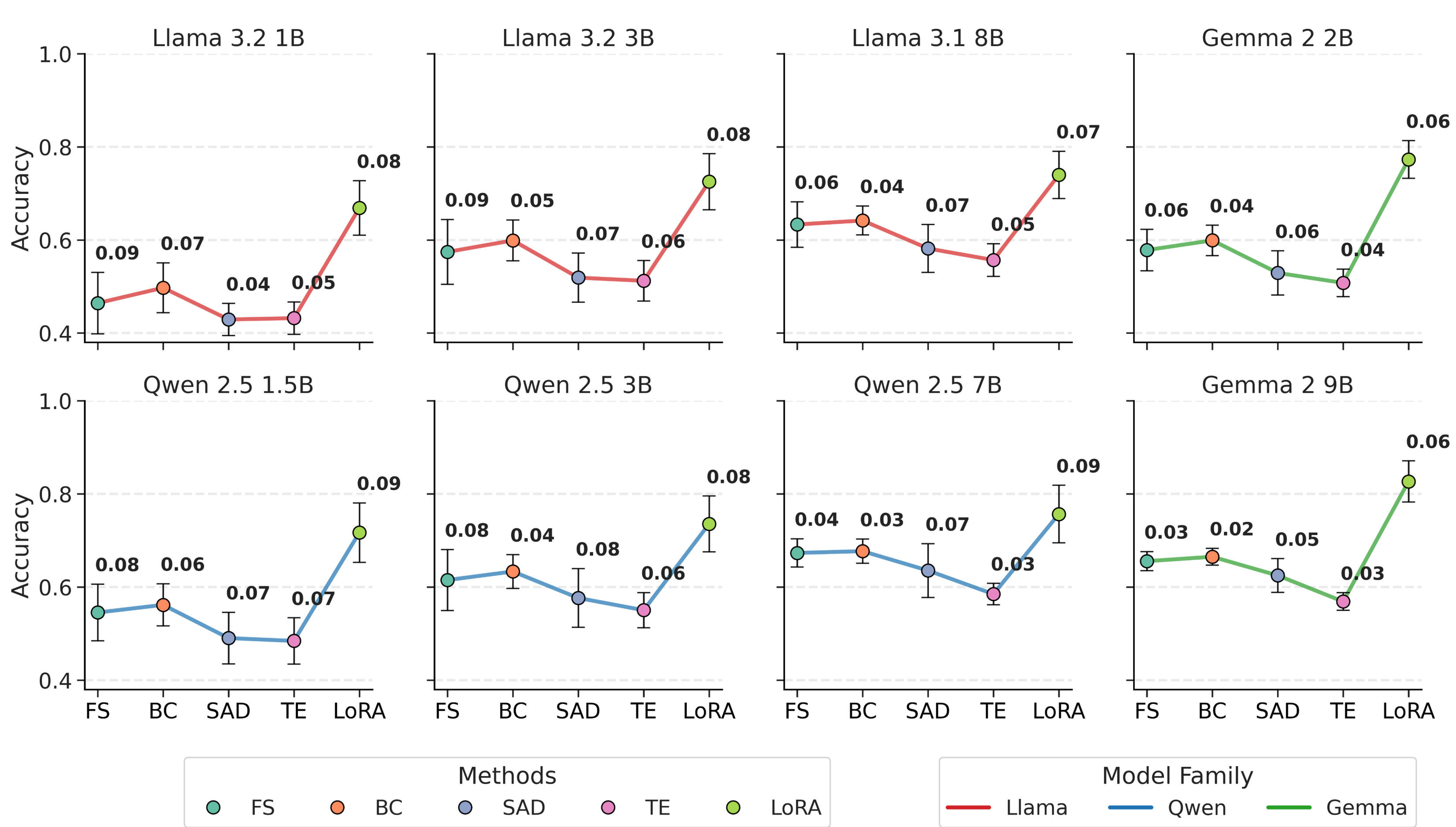
Formatting: we vary capitalization, space symbols, option item style, etc.

Capitalization: title, separator: '-', option item style: 'A, B, C'	**Capitalization: upper, separator: ':', option item style: '1, 2, 3'**
Question - {} A) {} B) {} Answer - {}	QUESTION: {} 1) {} 2) {} ANSWER: {}

RQ1: How robustness methods compare in efficiency in unified setting?

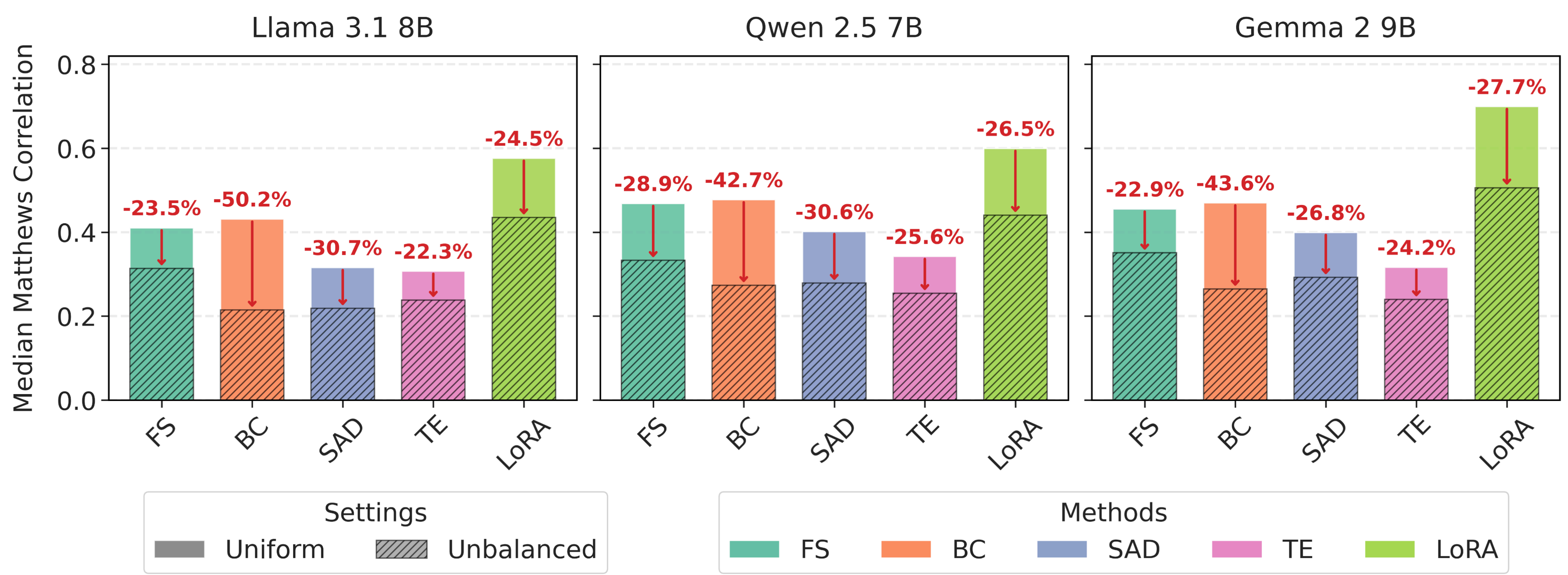
- **Batch Calibration** improves both accuracy and robustness.
- **Template Ensembles** reduce sensitivity but drop accuracy.
- **LoRA** improves accuracy but not robustness.

Action: use BC for uniform improvement over few-shot.



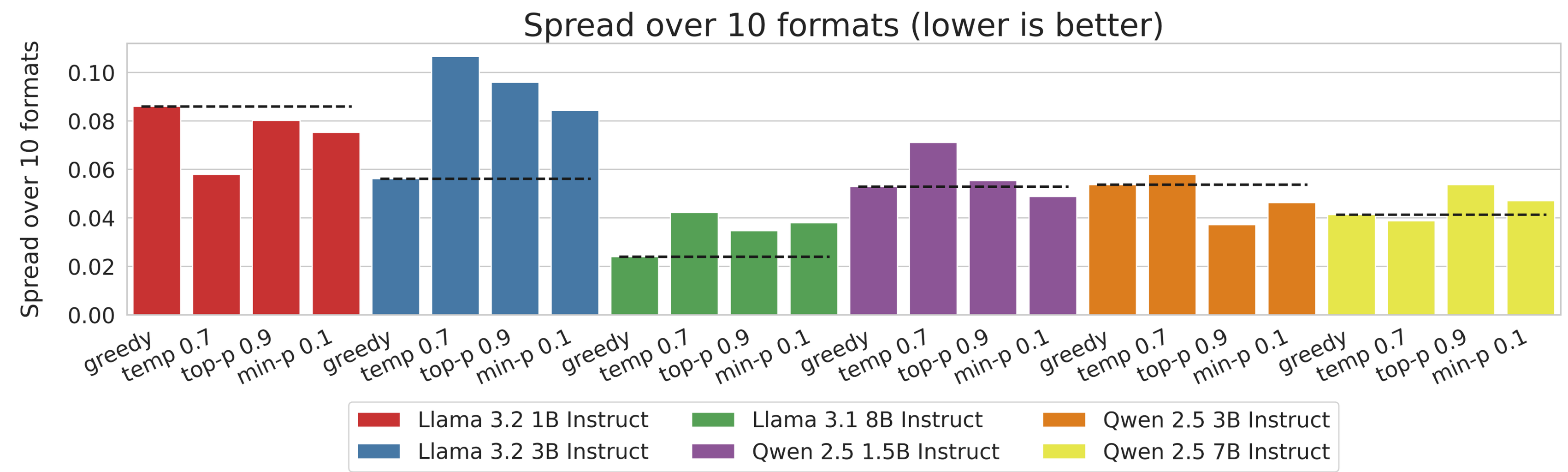
[1] Zhou et al., Batch Calibration: Rethinking Calibration for In-Context Learning and Prompt Engineering., 2024
[2] Voronov et al., Mind your format: Towards consistent evaluation of in-context learning improvements., 2024
[3] Lu et al., How are Prompts Different in Terms of Sensitivity?, 2024
[4] Qiang et al., Prompt perturbation consistency learning for robust language models., 2024

RQ2: How distribution shifts affect sensitivity of SFT and ICL-based methods?



BC is greatly affected by class imbalance.
During calibration scores for frequently predicted classes are decreased, and scores for rarely predicted classes are increased ⇒ BC has an implicit bias towards uniform prediction.
Action: choose data-aware prior when using calibration.

RQ3: How sampling strategies affect format sensitivity?



Comparison of sampling strategies and their effect on robustness to prompt-format variations in **long-form generation** (GSM8K)

- **Classification & multiple choice tasks:** probability ranking is more stable than greedy decoding.
- **Long-form generation tasks (Figure above):** results are model-dependent, and one sampling strategy might be twice as sensitive as the other.

Action: search over sampling strategies for your application.

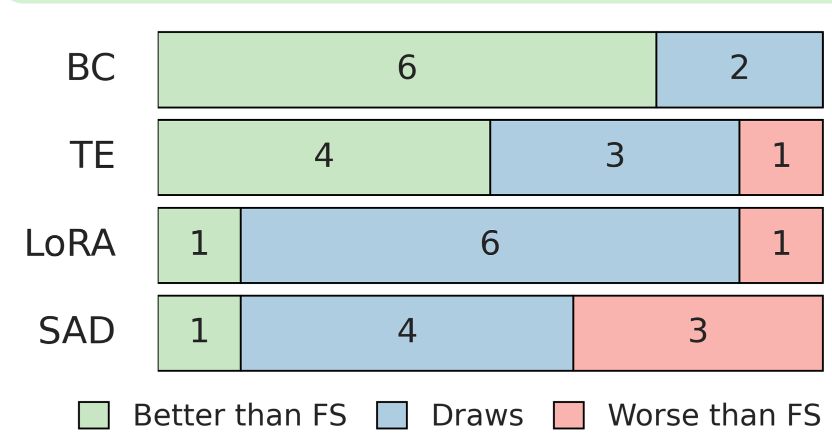
RQ4: How robust are frontier models?

Method	Model	Accuracy ↑	Std accuracy ↓	Spread ↓
Few-shot	Llama 3.1 8B	0.563	0.052	0.161
	Qwen 2.5 7B	0.605	0.058	0.190
	DeepSeek V3 0324	**0.741**	0.015	0.045
	GPT-4.1	0.624	**0.010**	**0.032**
Template Ensembles (majority voting)	DeepSeek V3 0324	**0.742**	0.009	0.028
	GPT-4.1	0.625	**0.005**	**0.018**

- **Frontier models are more stable than small models**
- Yet, some tasks still show 8–10 pt accuracy spread.
- We adapt Template Ensembles with majority voting instead of mean averaging, which reduces spread on 19/20 tasks (>44% reduction in 9) — likely because mode aggregation is more robust to outliers.

Action: use TE with majority voting if robustness is critical.

Appendix



Statistical test for spread reduction compared to the FS baseline.